Impact Analysis of Erroneous Data on IoT Reliability

2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation

Impact Analysis of Erroneous Data on IoT Reliability

Samuel J. Moore
School of Computing
Ulster University
Jordanstown, Northern Ireland
moore-s34@ulster.ac.uk

Chris D. Nugent
School of Computing
Ulster University
Jordanstown, Northern Ireland
cd.nugent@ulster.ac.uk

Ian Cleland
School of Computing
Ulster University
Jordanstown, Northern Ireland
i.cleland@ulster.ac.uk

Shuai Zhang
School of Computing
Ulster University
Jordanstown, Northern Ireland
s.zhang@ulster.ac.uk

***Abstract*—The ability to sense the environment is the cornerstone of the Internet of Things (IoT), which is a rapidly expanding paradigm that is altering the way we interact with machines. IoT enables a range of new services to enhance the lives of end-users. One of these services concerns activity recognition within Ambient Assisted Living which can be used to help people live independently at home for longer. Many of these applications can, however, be prone to failure and vulnerable to attack. Extensive research is therefore required to build towards a secure and sustainable IoT. This work examines activity recognition in a smart home environment using three different classifiers on a well-known activity recognition dataset. Fail-dirty and device shut-down data is introduced in the dataset to examine the impact that this erroneous data has on the application. This study found that it was possible to rank the importance of sensors with regards to their influence on classification by observing how these failures impacted the classifiers when compared to the f-measure produced from the classification of the clean data. This work also found that while representing data in a binary format obtains higher accuracy, it makes the classifier considerably more vulnerable to dirty data. Lastly, this study found that decision tree classifiers have an inherent vulnerability when it comes to handling dirty data, resulting in a 24% reduction in performance versus the clean data, due to the structuring and placement of leaf nodes in the tree.**

***Index Terms*—IoT, activity recognition, smart homes, reliability, classification, machine learning, fail-dirty**

I. Introduction

The Internet of Things (IoT) is a rapidly evolving paradigm which is significantly changing how we interact with computers in the physical world. IoT has a broad range of applications such as home security [1], healthcare [2] and monitoring traffic in smart cities [3].

Ambient Assisted Living (AAL) is an application within the IoT that aims to support independent living through the use of activity monitoring and recognition. As such, the ability to correctly, quickly and reliably classify activities within this domain is essential to the success of the application [4]. In the field of AAL, IoT deployments would typically be smart-homes equipped with a large number of sensors which may become difficult for engineers and carers to manage, especially given the fact that sensors do not often have user-friendly alerts or interfaces that can alert users when a problem occurs. Moreover, when we consider the limited battery power associated with remote sensing it becomes essential to check sensors on a regular basis to ensure they are working as expected. This task would be time consuming with a large number of sensors involved. Being able to rank sensors, based on an understanding of their reliability and relative importance to the classification task would allow carers to prioritise which sensors needed to be checked and how often, resulting in potentially huge time savings and a fuller situational awareness of the system. This becomes even more essential when we consider that remote sensors have a tendency to fail-dirty [5], which is where a sensor appears to be operating normally, but is actually communicating anomalous data.

Machine learning models are capable of classifying human activity based upon a given input [6]. This input is generally taken from sensor readings which may be placed in a smart-home environment or worn on the body. As such, the success or failure of the activity recognition model is highly dependent upon the sensors functioning correctly.

Reliability within IoT applications is a key area for research [7] due to the notion that IoT networks typically involve highly constrained devices [8], [9] communicating with eachother over lossy links [10]. The constrained nature of these devices make the IoT network considerably more vulnerable to device failure and security threats, and the growing frequency of these issues often leads to reduced trust by end-users [5]. With the issues regarding trust, security and reliability in mind, it therefore becomes essential that we build an awareness of the quality of our IoT systems. Quality and reliability are urgent requirements for IoT systems [11], [12] if we are to be able to fully integrate this technology into our everyday lives.

This research aims to examine the impact of two different and pervasive types of failure in IoT environments; fail-dirty and device shut-down failures. These two failures are simulated into a well-known activity recognition dataset to allow us to examine the impact of these anomalies. This analysis allows us then to draw conclusions around which sensors are most vulnerable to error, and the impact that they can have on the overall classification performance, across all classes in the model. This study performs this failure analysis on two different data preparation approaches; binary and numeric representation, to determine if either of these two approaches

This research is supported by the BTIIC (British Telecom Ireland Innovation Centre) project, funded by BT and Invest Northern Ireland.

978-1-7281-4034-6/19/$31.00 ©2019 IEEE
DOI 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00335

are more susceptible to failure. Three different classifiers were trained and tested in this study for all cases being studied; a binary and numeric representation of each classifier, and then within each of these the two different failure types were introduced across the 14 sensors, resulting in a total of 168 tests, the main findings of which are discussed in this paper.

The remainder of this paper is organised as follows: Section II is a literature review of IoT issues, activity recognition, and data preparation for classification. Section III provides detail on the methodology used for this experiment. Section IV details the findings from the experiment and discusses the impact of these on the wider field of research. Section V discusses opportunities for further research, and Section VI concludes this study.

II. Related Work

Research is continuing to grow in the fields of both IoT reliability and activity recognition. The increasing availability of low cost sensor and communication technology is enabling us to create connected spaces that open up new opportunities. AAL generally concerns a smart environment, equipped with sensing capability that allows us to infer activity. Chen et al. [4] defined this paradigm as "dense sensing", where the environment is embedded with a large number of low-cost, low-power miniature sensors. These sensors are normally embedded into objects, which the human will then interact with. These simple human-object interactions can provide valuable information pointing to the activity being undertaken.

Jurek et al. [13], examined activity recognition using ensemble classifiers. The study provides deep level detail on the data preparation process needed to successfully infer activity information from a feature vector generated in a smart-home environment. The study describes two fundamental ways of representing feature vectors; numerically and binary. A feature vector is represented as:

$$S = (S_0, S_1 ... S_n)$$

where S is a sensor and n is the number of sensors in that feature vector. The numeric representation of the feature vector would mean that the range of S_i is:

$$S_i = [0, 1...n]$$

where n represents the number of times that S_i was fired during the window of time represented by the feature vector. The binary representation of the feature vector would mean that the range of S_i is:

$$S_i = [0, 1]$$

where S_i has only two possible values within the feature vector. A zero indicates that the sensor did not fire during the time window, and a one indicates that the sensor fired at least once during the window.

Interestingly, this study found that the binary representation enabled the classifier to have a higher performance, resulting in a unanimous improvement on classification f-measure. In the areas for further research in this paper the author indicates that further research is required to understand these two representations for classification to ascertain if the number of times the sensor was triggered may be significant with respect to handling anomalous data.

IoT networks are known to be vulnerable to hacking attempts. An example of constrained IoT devices being exploited made mainstream news when hackers leveraged connected surveillance cameras to bring down an entire network [14]. This problem is front-of-mind for many legislative bodies as the public and private sector quickly attempt to secure the IoT, which is evidenced by the U.K. government producing consumer guidelines for the production of smart objects in November 2018 [15].

In combination with IoT's well documented security vulnerabilities there are also some concerning data quality characteristics related to IoT. These characteristics are described in detail in [5]. One of the concerning characteristics documented in this paper is the constrained nature of the devices in terms of power, battery and storage. These constraints limit the devices ability to perform complex operations, such as cryptography. They also tend to operate on battery, which leads to a concern where we are not always aware of the status of the battery, meaning that the device could fail at any time without warning.

Another concerning characteristic which is detailed in [5] is the propensity for IoT sensors to "fail-dirty", which is a particularly concerning phenomenon. This type of failure, which comes without warning and is pervasive in IoT environments, is a cause for concern - especially in circumstances where IoT applications have a direct impact on humans, such as AAL.

The author of [5] also describes IoT applications' tendency to drop sensor readings. Depending on the quality of service (QoS) standards of the protocols in use, which in IoT applications are heterogeneous and varied, there may not be delivery guarantees associated with data transmissions meaning that data can be dropped with no warning.

The tendency for IoT applications to lose power, drop readings, and fail-dirty points to a prudent research question: do we understand the impact that anomalous data has on an IoT application? While the issue of fail-dirty data is well documented in the literature, there is a lack of literature that observes the impact of this erroneous data. Without first understanding the impact of these issues, it is not possible to fully understand the problem domain. This study is a novel contribution to the literature, through its analysis of the impact of common failures in a typical IoT environment. This study serves as an important first step in determining the overall reliability of our IoT systems.

III. Methodology & Dataset

This experiment seeks to assess the impact of fail-dirty and device failure (i.e., loss of battery power) on the performance of an activity recognition classifier in an IoT environment. This experiment will introduce these two types of failures into two different representations of the data, binary and numeric. The remainder of this Section provides the methodology for the experiment.

A. Dataset Selection

A well-known dataset for activity recognition was identified for use in this experiment, the details of which are discussed fully in [16]. To summarise, the dataset consists of 14 digital state-change sensors that were deployed in the home of a 26-year-old male and collected data over a period of 28 days. During this time, the inhabitant wore a bluetooth headset, through which he annotated each activity as it occurred. This resulted in a total of 2,120 sensor events and 245 activities being recorded throughout the 28-day time period.

The 2,120 sensor events are recorded in a stream of data which details the time the sensor began firing and the time that it stopped. In order to transform the data into a state conducive to activity recognition it must undergo a windowing process. This process is discussed in detail in [13], and both the binary and numeric windowing approaches have been tested in this experiment.

B. Classifier Selection

Three classifiers were identified for use in the experiment; Naive Bayes, Decision Tree and a Neural Network. These classifiers were identified due to their popularity and suitability to the task of classification in an IoT environment, as discussed in [17] and [6]. Python's Scikit Learn library [18] was used to implement this experiment. With regards to the specific algorithms chosen from Scikit Learn's library, they were as follows: for the neural network, Multi-layer Perceptron classifier was used. For the decision tree, the Decision Tree Classifer was used. For Naive Bayes, the numeric representation was trained and tested with the Multinomial Naive Bayes algorithm, while the binary representation was trained and tested with the Bernoulli Naive Bayes algorithm. The reason for requiring a separate classifier for Naive Bayes is because one of the algorithms (Multinomial) is designed for continuous data and therefore is not suited to binary data, so it was biased towards the numeric data. Bernoulli is designed for binary data, but cannot be used on the numeric data because it would transform each feature vector into binary representation, therefore making the results identical to the binary results.

C. Data Preparation

In order to measure the impact of failures and anomalous data, a baseline must be established so that the extent of the failures can be benchmarked against it. This baseline was created by training the three classifiers on a clean version of the binary and numeric data. F-measures would then be recorded from the tests performed against this data. Therefore, once the trained classifiers were given the anomalous data to test, we can easily observe the impact that this anomalous data has on the classifier by observing how it changes the F-measure.

The dataset was divided into a train and test set using the train_test_split method from Scikit Learn, this allows for a set of clean training data to be segregated for use in the training of the classifiers, meaning that the test data will not have been seen by the classifier in training. The same seed was used in all cases for all classifiers to ensure that each classifier was trained on the same training cases, and then each classifier would be evaluated against the same test cases meaning that the results would be a fair comparison across the classifiers. The proportion decided for the training and test sets was 50%. This number was chosen based upon van Kasteren's analysis in [16] that increasing the training data beyond half does not yield higher accuracies.

D. Simulation of Anomalies

A variety of errors were introduced into the data so that the F-measure for the classification performance could be measured when the classifier was fed dirty data. These errors were categorised as follows:

- Simulation of device power failure - changing all sensor readings to zero for a given sensor.
- Simulation of fail-dirty data - inserting false sensor readings into a feature vector for a given sensor.

In the case of the binary data, this is achieved by inserting a value of one or zero. For the numeric data, this has been tested by inserting zero, one, median and max values into each separate sensor, performing an individual test with anomalous data for each sensor.

Once failures have been simulated, the f-measure will be collected for each test and compared against the clean f-measure. Impact will be assessed by a gap analysis between the clean and dirty f measures.

IV. Results and Discussion

Using the methodology described in the previous Section, results were produced for the three classifiers. Firstly, the baseline f-measures were established for each classifier, which are described in subsection IV-A. Next, results were generated for fail-dirty and device shut-down simulations across each of the 14 sensors in the environment, this was produced on both the numeric and binary representations of the data and for all three classifiers, meaning that a total of 168 result sets were produced. The main findings of these results are presented and discussed in this Section.

A. Baseline Performances of Classifiers

The trained classifiers for Naive Bayes, Decision Tree and Neural network achieved f-measures of 96.1%, 94.1% and 100.0%, respectively when trained and tested on the clean binary data, as shown in figure 1. Using the numeric representation of the data, these metrics were 92.7%, 93.1% and 100.0%, respectively. Figure 1 illustrates the f-measures collected from the classifiers when the two types of failure were introduced, shut-down and fail-dirty, on both the binary and numeric datasets. The f-measure, in this case, represents a high-level view of the overall classification performance, without providing individual detail on how each class within the classifier performed. Even at this high level we can already see some themes emerging. Firstly, we see that the most largest reductions from the clean f-measure, after failures are introduced, are found in the Decision Tree classifier - this is

discussed in more detail later in this section. Secondly, we see that while the use of binary data increases the f-measure when using clean data, there are large reductions from the clean f-measure when errors are introduced into the binary data. Moreover, these reductions are considerably more significant in the binary data than they are in the numeric representation.

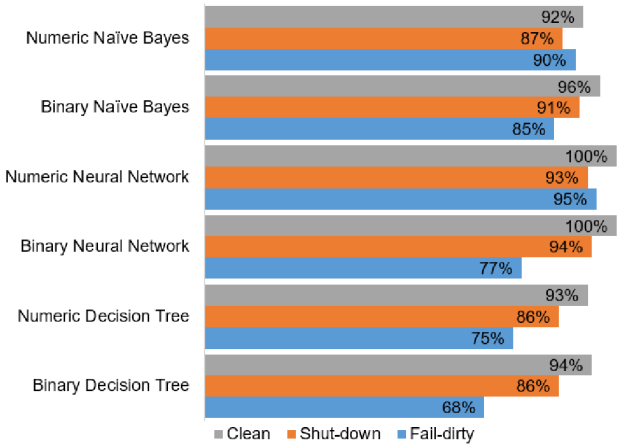

Fig. 1. F-measure analysis of classifiers on numeric and binary datasets

B. Binary vs. Numeric Data Performance

Almost unanimously, across all classifiers and sensors tested, there was a larger gap in f-measure performance when using binary data, rather than the numeric data. Figure 2 shows an analysis of the neural network results for both numeric and binary shut-down and fail-dirty data to illustrate this point, and the same thing can be seen in the naive bayes and decision tree results. The numbers and bars in figure 2 represent the reduction in total f-measure observed from the clean performance after dirty data was introduced. There is a column for binary and numeric for both shut-down failures and fail-dirty data. Firstly, we can observe that the reductions are much larger in the binary cases. Secondly, we can observe that within the binary failures, the largest reduction is to be seen with the fail-dirty data. Given these results, we can conclude that when we reduce the complexity of the data we make the classifier significantly more vulnerable to erroneous data, in particular fail-dirty data. Therefore, an IoT system architect is left with a difficult decision, given that there is higher accuracy to be gained from the binary representation, but pursuing this avenue opens up vulnerability to failure.

This problem is illustrated further when we compare the sensor failure data with human reasoning of which sensors might cause the biggest drop in f-measure. For example, from a human reasoning perspective, if we examine a feature vector and see that the toilet flush sensor has been triggered, we would likely conclude that the inhabitant had used the toilet, so in the case of a fail-dirty toilet flush sensor we would often be led to mislabel activities based on this fail-dirty sensor informing us that the toilet flush sensor is firing when it is

Sensor	Binary Shut-down	Numeric Shut-down	Binary Fail-dirty	Numeric Fail-dirty
1-Microwave	3%	3%	20%	4%
2-Hall-Toilet-door	15%	15%	0%	0%
3-Hall-Bathroom-door	3%	2%	37%	4%
4-Cups-cupboard	0%	3%	41%	11%
5-Fridge	6%	6%	43%	3%
6-Plates-cupboard	4%	16%	36%	10%
7-Frontdoor	16%	16%	42%	3%
8-Dishwasher	0%	0%	10%	4%
9-ToiletFlush	3%	4%	27%	8%
10-Freezer	9%	7%	15%	7%
11-Pans-Cupboard	4%	9%	14%	9%
12-Washingmachine	0%	0%	2%	0%
13-Groceries-Cupboard	4%	0%	8%	3%
14-Hall-Bedroom-door	17%	17%	25%	10%

Fig. 2. Failure impacts on numeric and binary datasets using neural network classifier

not. Nonetheless, we can observe from figure 3, which is the confusion matrix for a fail-dirty toilet flush sensor on the numeric representation of the naive bayes classifier, that only a single activity instance is misclassified as ”use-toilet”.

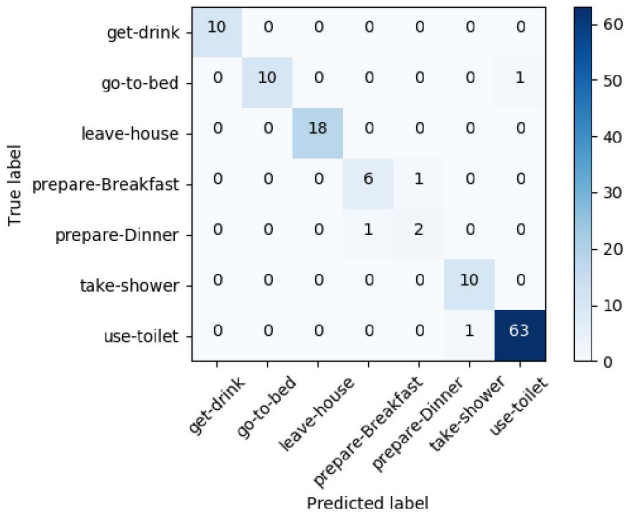

Fig. 3. Confusion matrix for fail-dirty toilet flush sensor using Naive Bayes on numeric data.

This is contrasted greatly when we examine the same failure simulated on the binary data. Using the binary data and classifier, we see that the f-measure falls by 17% which compared to the 0% drop in accuracy using the numeric data is a large reduction by comparison. Figure 4 is the confusion matrix for the fail-dirty toilet flush sensor on the binary Naive Bayes classifier, showing that a total of 22 activities from 3 different classes were mislabelled as ”use-toilet”. This reduction of performance points to a very serious concern regarding the use of binary data for this classification task. Often, care providers cannot run the risk of critical activities being mislabelled due to one sensor transmitting faulty data.

The problem can be extrapolated further when we con-

sider the inherently insecure arena of IoT, and the possibility of hacking attempts mentioned earlier in this paper. The consequences would be dire if a malicious individual staged a man-in-the-middle attack on this sensor network to deliberately mislead the classifier when we consider the vulnerable inhabitants that these systems are serving. With this in mind, engineers must make careful and considered decisions when choosing between binary and numerically trained classifiers for activity recognition. Binary classifiers do increase performance, but open up vulnerabilities which may not be worth the risk.

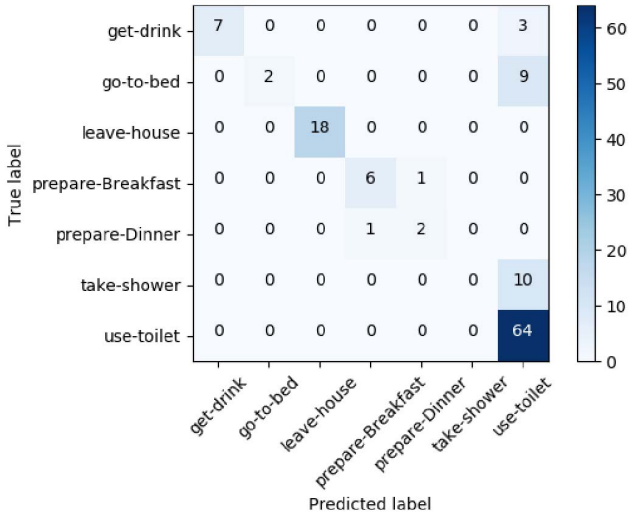

Fig. 4. Confusion matrix for fail-dirty toilet flush sensor using Naive Bayes on binary data.

C. Sensor Prioritisation

A key objective of this study was to determine the possibility of developing a ranking of sensors in the environment in order to gain an understanding of the operational quality of the system should a given sensor fail. By analysing the impact that the failure of a single sensor has on the overall classification performance, we can begin to form a rudimentary ranking of sensor importance.

Figure 5 illustrates failure impacts of both device shut-down from power failure and the introduction of fail-dirty data on the numeric dataset with the Naive Bayes classifier. From this we can observe that some sensors have a more significant impact on the f-measure when erroneous data is introduced. The highest impact from a single error-type on a single sensor is to be found on the front door sensor when device shut-down occurs. The confusion matrix for this particular failure is presented in Figure 6, and this shows that for the 18 activities labelled "leave-house", the NB classifier was unable to classify any correctly. By contrast, the NB classifier correctly classified 100% of activities labelled "leave-house" when tested on clean data. This finding illustrates that the sensor on the front door is vital to the classification of one activity, and when the sensor fails it becomes impossible to correctly classify the activity. This finding was also observed in both the neural network and decision tree experiments. In environments where ADLs are being classified to monitor patient health, the activity of leaving the house is critical. Consider the example of a dementia patient: the care staff may be relying on an alert being triggered by the classification of this activity, and a single sensor failure could jeopardise this entirely. From this we can begin to extrapolate a list of which sensors are most critical to the environment based upon those which have the highest impact on classification.

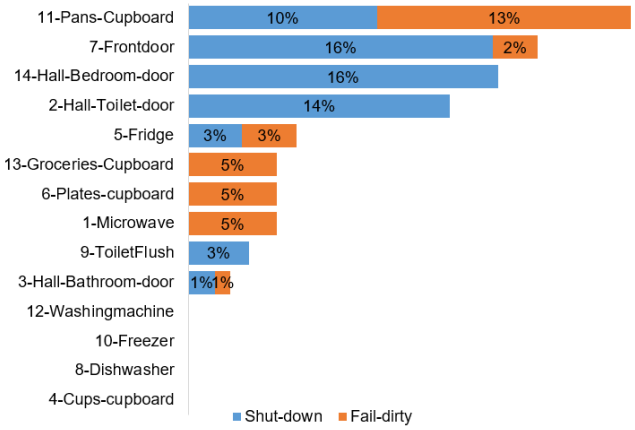

Fig. 5. Impact of failure by sensor on Naive Bayes

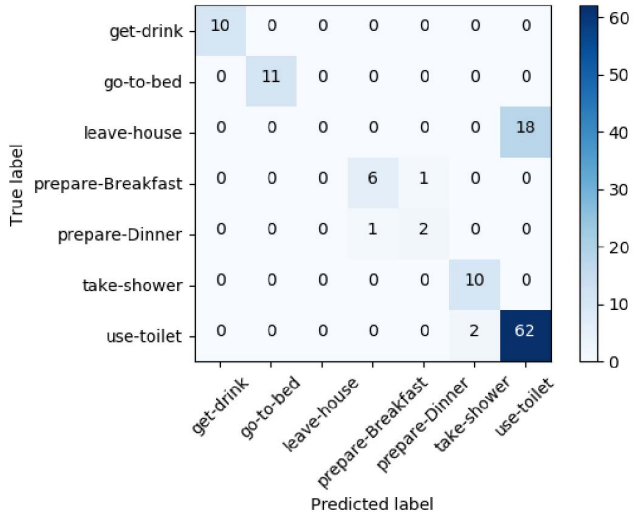

Fig. 6. Naive Bayes confusion matrix for device shut-down on front door

Feature selection is often used in machine learning to identify the features that contribute the most to classification in order to reduce the dimensionality of the data. In particular, Chi-square is well suited to multi-class problems, as opposed to other feature selection methods [19]. A comparative analysis between chi-square, and the failure impacts from

this experiment was produced to ascertain if a correlation was found between the two approaches to further cement the understanding of individual sensor importance within this deployment, results of this are shown in 7.

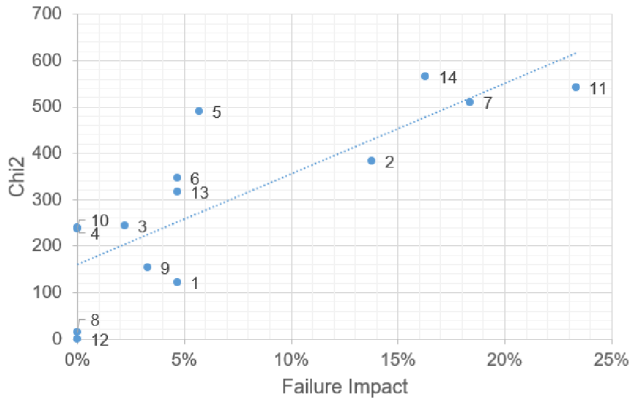

Fig. 7. Scatter graph plotting the chi-square results against the failure impact scores using the numeric naive bayes data. r=0.8097

Using the scatter graph from Figure 7, we can ascertain that there is a strong positive correlation between the chi-square results and the analysis of the failure impact scores, resulting in an correlation coefficient of 0.8097. The top right quadrant of the graph represents the most critical sensors in the environment. This verifies that should these sensors fail, we would see a large reduction in the performance of the classifier. As such, these sensors should be treated as the highest priority within the environment. This methodology of determining sensor priority could be applied to any generic sensor environment, allowing IoT architects to build a strong situational awareness of the IoT deployment with regards to information reliability.

D. Resilience of Decision Trees to Device Failure

Earlier in this Section, it was illustrated that the decision tree classifier had notably lower f-measures once erroneous data was introduced. The f-measure for the decision tree when using clean data was 94.1%, but this metric was reduced significantly when tested with device shut-down and fail-dirty data, scoring 86.4% and 67.8%, respectively. This reduction in accuracy is a serious cause for concern that, given the inherently insecure IoT applications at hand, would indicate that a decision tree is not a suitable classifier for activity recognition applications in the real world.

Figure 8 depicts the individual impact scores for fail-dirty and shut-down errors for each sensor in the deployment for the decision tree. We can observe here that some sensors have a disproportionately large impact on the classifier, whereas other sensors have no impact at all. A fail-dirty front door sensor results in a 75% reduction to the f measure, whereas failures to the washing machine, cups and microwave sensors exhibited no impact at all during testing. This suggests a concerning

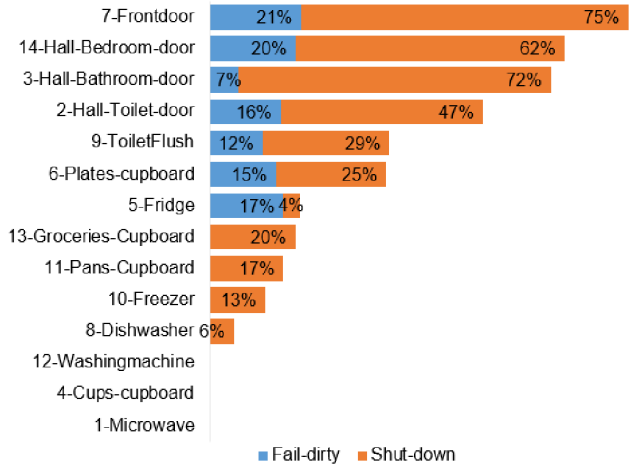

Fig. 8. Failure impacts on decision tree based on binary data

behaviour with decision tree classifiers when handling these types of errors.

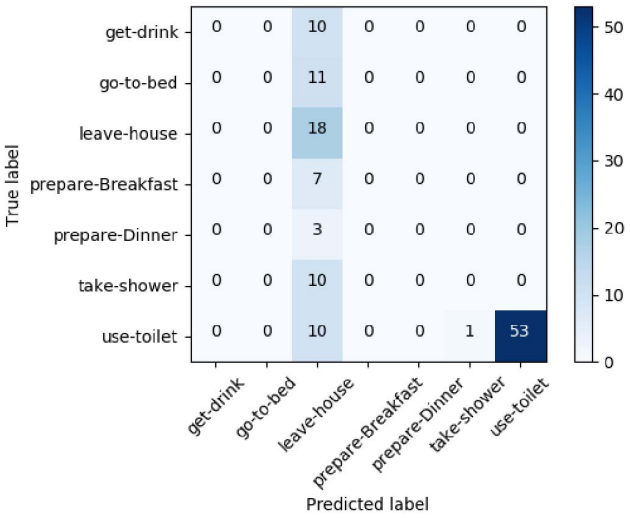

Fig. 9. Confusion matrix for fail-dirty door sensor using binary data

Figure 9 presents the confusion matrix for the fail-dirty front door sensor. We can observe that the majority of instances have incorrectly been classified as the activity ”leave-house”. This makes sense, given that the dirty data has been simulated on the front door sensor. As discussed earlier in this section, we know the front door sensor is vital with regard to classifying the leave-house activity. Perhaps what is most concerning here is the impact that the fail-dirty door sensor exhibits on other labels which typically do not rely on the door sensor for classification. By contrast, the Neural Network and Naive Bayes were over 30% more accurate with this particular failure, which illustrates that in this scenario the weakness resides in the decision tree classifier, rather than the sensor.

Decision trees are structured by a series of leaf nodes,

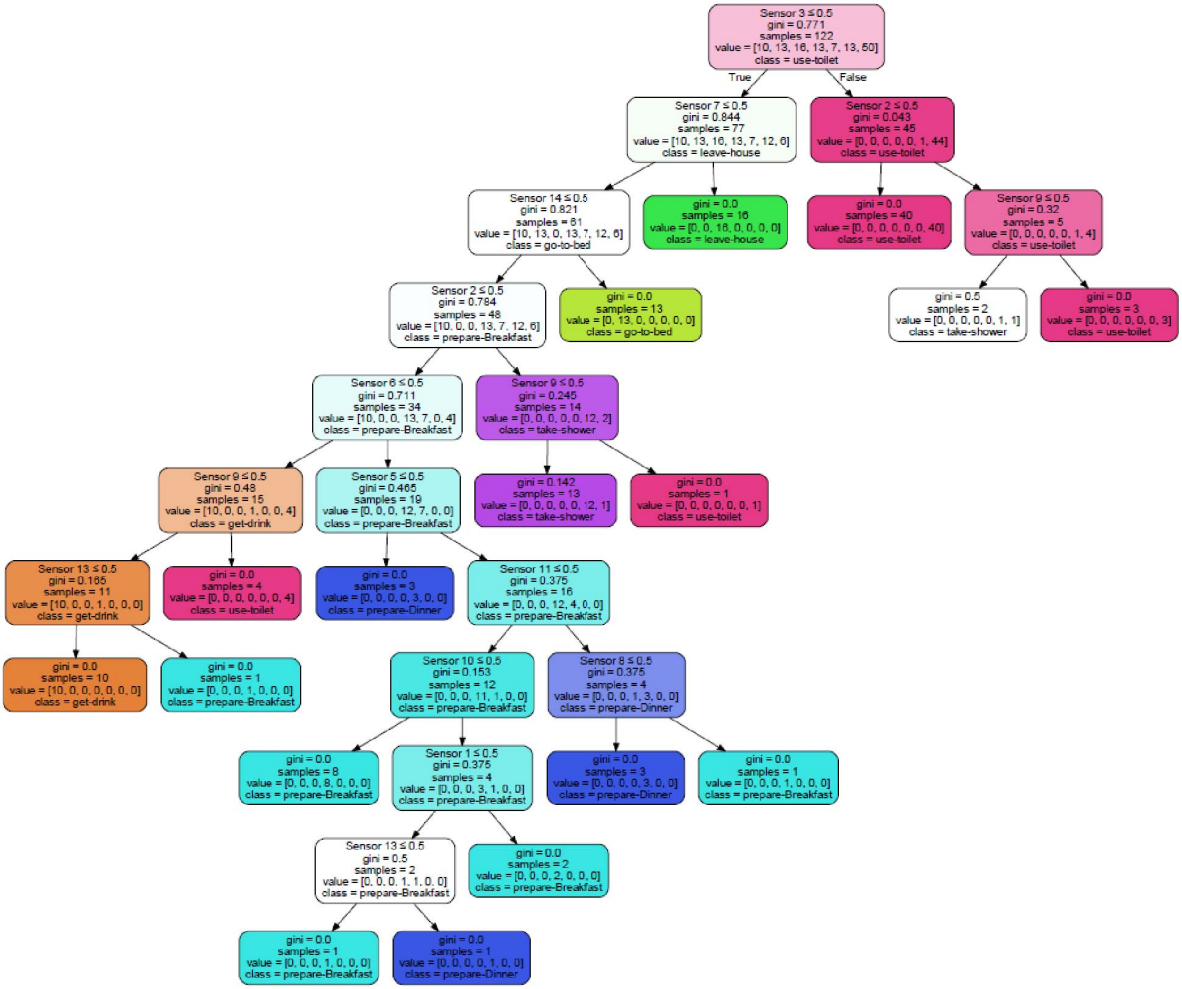

Fig. 10. Structure of the decision tree classifier

each of these nodes representing a decision boundary for a feature within the dataset. The decision tree structures the leaves hierarchically and begins at the top of the tree then works down to the leaf nodes at the bottom. If a threshold is met that satisfies the tree's learned boundary of that class, then the tree will classify the activity and move on to the next. Importantly, the decision tree does not always reach the lower-most leaf node. For example, depending on the structure of the data and its dimensions, if a threshold is met on the first leaf node at the top of the tree it is possible for the tree to classify that activity immediately, and therefore it would not examine the other thresholds at the lower leaf nodes. Figure 10 shows the structure that was generated by the Scikit Learn decision tree classifier algorithm. This structure first examines the threshold for sensor 3, checking if there is a 1 or a 0 reading for the sensor. If sensor 3 is reading as a 0, then the next leaf node checks sensor 7 (which in a fail-dirty scenario is going to be entirely populated with 1 values) to gauge if the value is a 1 or a 0. If the value is a 1, then the activity is classified as leaving the house. The decision tree makes this decision without full awareness of the values in the other 13 sensors. With this particular tree structure, the only way to avoid misclassification of activities in the event of a fail-dirty front door sensor is if the decision tree never reaches that leaf node, which is unlikely considering how high the leaf node is in the tree's hierarchy. This explains why the "use-toilet" class was largely unaffected by the fail-dirty door sensor, because if sensor 3, the bathroom door, is triggered then the leaf node for sensor 7 is avoided. Unfortunately, with this tree, the vast majority of leaf nodes can only be reached by passing through sensor 7, meaning that all activities except for using the bathroom are entirely jeopardised by one fail-dirty sensor.

With the results of this decision tree classifier in mind, we must critically examine the role of decision tree classifiers within activity recognition IoT environments. By using a decision tree, we are leaving the accuracy of the model at the mercy of the tree structure, which due to its hierarchical nature,

is destined to fail in the event of erroneous sensor readings. Perhaps most concerning is the fact that, on a clean set of data, the decision tree classifier performs extremely well: this could potentially mislead developers into a false sense of security. When using any classifier we must consider how resilient it is to failure, and this experiment demonstrates that the decision tree, while adequate on clean data, is simply not robust when it comes to handling dirty data. Given the tendency for constrained IoT environments to experience failure this is of paramount importance.

V. Areas for Further Research

This study has identified some key concerns within the arena of IoT data quality and reliability. With respect to the tendancy for IoT sensors to fail-dirty, there is a need to observe this phenomenon as it naturally occurs in a real-world dataset. Given the nature of how this error occurs will mean it will be challenging to catch, but it would be essential for researchers to fully understand the phenomenon.

This study has also identified the impact of given failures in an IoT environment, the logical next step would be to identify some pre-emptive measures which identify these failures in real-time so that they can be alerted before they are fed into the classifiers. One possible way of doing this, given the Markovian nature of the problem, would be to use a Markov Chain to analyse the probability of a given state transition to identify erroneous patterns in the raw sensor data.

VI. Conclusion

This study examined the impact of device shut-down and fail-dirty data on a well-known activity recognition dataset [16] across three different classifiers; Naive Bayes, Decision Tree and Neural Networks in both numeric and binary representations of the feature vectors. The study found that, while performance of the classifier is enhanced when operating on clean binary data, there are concerning impacts to all binary classifiers in the study when sensors transmit erroneous data, making the application much less reliable.

The study was also able to identify a group of sensors that had the most significant impact on classification through a fusion of the chi-square feature selection method and the failure analysis of the fail-dirty and device shut-down data. This group of sensors can then be treated as high-priority within the environment and be given special care and attention by engineers and care-home staff, in order to mitigate against the possibility that they might fail and severely damage the application.

Lastly, this study unveiled a concerning characteristic of the decision tree classifier which illustrates the algorithms inability to handle erroneous fail-dirty data. Given the propensity for IoT applications to fail, or fall victim to attacks, this indicates that the decision tree classifier is fundamentally unsuitable to classification tasks within IoT environments.

References

[1] H. Reza Ghorbani and M. Hossein Ahmadzadegan. Security challenges in internet of things: survey. In *2017 IEEE Conference on Wireless Sensors (ICWiSe)*, pages 1–6. IEEE, nov 2017.

[2] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243, 2014.

[3] Dhananjay Singh, Gaurav Tripathi, and Antonio J. Jara. A survey of Internet-of-Things: Future vision, architecture, challenges and services. *2014 IEEE World Forum on Internet of Things, WF-IoT 2014*, pages 287–292, 2014.

[4] Liming Chen, Jesse Hoey, Chris D. Nugent, Diane J. Cook, and Zhiwen Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 42(6):790–808, 2012.

[5] Aimad Karkouch, Hajar Mousannif, Hassan Al Moatassime, and Thomas Noel. Data quality in internet of things: A state-of-the-art survey. *Journal of Network and Computer Applications*, 73:57–81, 2016.

[6] Uday Shankar Shanthamallu, Andreas Spanias, Cihan Tepedelenlioglu, and Mike Stanley. A brief survey of machine learning methods and their sensor and IoT applications. In *2017 8th International Conference on Information, Intelligence, Systems and Applications, IISA 2017*, volume 2018-Janua, pages 1–8. IEEE, aug 2018.

[7] Narendra K. Saini. Trust factor and reliability-over-a-period-of-time as key differentiators in IoT enabled services. *2016 International Conference on Internet of Things and Applications, IOTA 2016*, pages 411–414, 2016.

[8] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys and Tutorials*, 17(4):2347–2376, 2015.

[9] Ammar Rayes and Samer Salam. *Internet of things-from hype to reality: The road to digitization*. 2016.

[10] Berihun Fekade, Taras Maksymyuk, Maryan Kyryk, and Minho Jo. Probabilistic Recovery of Incomplete Sensed Data in IoT. *IEEE Internet of Things Journal*, 5(4):2282–2292, 2017.

[11] Sabrina Sicari, Alessandra Rizzardi, Daniele Miorandi, Cinzia Cappiello, and Alberto Coen-Porisini. A secure and quality-aware prototypical architecture for the Internet of Things. *Information Systems*, 58:43–55, jun 2016.

[12] Tom Arjannikov, Simon Diemert, Sudhakar Ganti, Chloe Lampman, and Edward C. Wiebe. Using Markov Chains to Model Sensor Network Reliability. *Proceedings of the 12th International Conference on Availability, Reliability and Security - ARES '17*, pages 1–10, 2017.

[13] Anna Jurek, Chris Nugent, Yaxin Bi, and Shengli Wu. Clustering-based ensemble learning for activity recognition in smart homes. *Sensors (Switzerland)*, 14(7):12285–12304, 2014.

[14] LILY HAY NEWMAN. Friday's East Coast Internet Outage Is a Major DDOS Attack — WIRED, 2016.

[15] DDCMS. Code of Practice for Consumer IoT Security. (October), 2018.

[16] Tim van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*, page 1, 2008.

[17] Nesma Settouti, Mohammed El Amine Bechar, and Mohammed Amine Chikh. Statistical Comparisons of the Top 10 Algorithms in Data Mining for Classi cation Task. *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(1):46, 2016.

[18] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake Vanderplas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[19] Ikram Sumaiya Thaseen and Cherukuri Aswani Kumar. Intrusion detection model using fusion of chi-square feature selection and multi class SVM, 2017.